DailyHW\_Day4\_Week6

03/31/2022

Notes and Q&A:

<https://docs.google.com/document/d/1O7ogrWUuttxBATKADPk98zPqyQcgCWbdF18YKx0xcSQ/edit?usp=sharing>

Checklist:

<https://docs.google.com/document/d/1fmYhFHUIy44UIId1fpUUV-vWt9RdjD0ghkv9n5PqzBw/edit?usp=sharing>

1. What have you reviewed today?

**Spring**

* Spring framework provides:
  + Managing objects and dependencies by application context and dependency injection
  + Data access
  + Spring mvc
* Dependency injection types:
  + Constructor
  + Setter
  + Field
* AOP: Solution for cross-cutting concerns(logging, security, transaction)
  + Aspect
  + JoinPoint
  + Advice
  + PointCut
* Propagation level(@transaction)
  + Required
  + Required\_new
  + Support
* Isolation level(@transaction)
  + 4

**SpringMVC**

Flow

* all incoming request is intercepted by the **dispatcherServlet**
* **dispatcherServlet** gets an entry of handler mapping form the xml file and forwards the request to the controller
* the controller returns an object of **ModelAndView**
* **dispatcherServlet** checks the entry of the **view resolver** in xml file and invokes the specified **view component**.

**Spring Boot**

* **Advantage:**
  + a flexible way to config the java beans, xml configuration and database transactions
  + batch processing and manage rest endpoints
  + everything is auto configed
  + annotation based spring application
  + eases dependency management
  + include embedded servlet container -> Tomcat

swagger is an open-source project used to generate the rest apiful documents for RESTful services.

1. What new things you have learned today?

Today, we dive into microservice architecture and actual coding.

Microservice architecture

* Management/orchestration
* responsible for placing services on nodes, identifying failures, rebalancing services across nodes, and so forth.
* Typically, this component is an off-the-shelf technology such as Kubernetes, rather than something custom built.
* API Gateway.
* The entry point for clients. Instead of calling services directly, clients call the API gateway, which forwards the call to the appropriate services on the back end.

Advantages of an API gateway:

* It decouples clients from services. Services can be versioned or refactored without needing to update all the clients.
* Services can use messaging protocols that are not web friendly, such as AMQP.
* The API Gateway can perform other cross-cutting functions such as authentication, logging, SSL termination, and load balancing.
* Out-of-the-box policies, like for throttling, caching, transformation, or validation.

1. What is your learning plan tomorrow?

* 2 coding questions
* Review what have been covered this week